

UNIVERSIDADE DE LISBOA  
FACULDADE DE CIÊNCIAS  
DEPARTAMENTO DE ESTATÍSTICA E INVESTIGAÇÃO OPERACIONAL



## **Rotas de recolha de produtos num armazém e rotas de distribuição de produtos numa zona de clientes**

João Miguel dos Santos Araújo

**Mestrado em Estatística e Investigação Operacional**  
Especialização em Investigação Operacional

Relatório de Estágio orientado por:  
Professora Maria da Conceição da Fonseca  
Dr. Carlos José Lúcio Martins



Este estágio foi realizado na empresa Norparts uma das empresas da Create Business. Agradeço ao Dr. Carlos Martins da Norparts esta oportunidade e todo o apoio prestado.



## **Agradecimentos**

Inicialmente, quando pensei em fazer a tese, olhava para outras teses e achava sempre que a página de agradecimentos era algo que não fazia muito sentido. Agora que estou a concluir este projeto e depois de todo o percurso realizado, percebo finalmente a razão e importância desta página.

Quero agradecer em primeiro lugar à pessoa que considero ter sido a mais importante para a realização deste trabalho, a professora Maria da Conceição da Fonseca. Houve momentos de grande dificuldade durante a realização desta tese e sinto que a ajuda da professora foi essencial. Tanto no aspeto da realização da tese em si, como na vontade e motivação para continuar a trabalhar. Seja dentro ou fora do estágio, senti sempre uma grande disponibilidade por parte da professora e penso que o mínimo que posso fazer é agradecer-lhe.

Em segundo lugar, quero agradecer à pessoa que mais me acompanhou durante o mestrado e ainda um pouco durante o final da licenciatura, a minha colega de curso Laura Pinguicha. O acompanhamento começou no final da licenciatura quando me juntei com a Laura para fazer o trabalho de uma das cadeiras. A partir daí, sempre que havia um trabalho de grupo, a Laura foi sempre a minha parceira. Trabalhar com a Laura foi sempre algo que me agradou bastante, não só porque é uma excelente aluna, mas também porque para além de me aturar imenso, ainda ajudou a despertar o espírito crítico e a vontade de fazer melhor.

Em terceiro lugar, quero agradecer à segunda pessoa que mais me acompanhou durante o mestrado, a minha colega de curso Inês Coelho. Desde o início do mestrado que a Inês sempre foi uma das minhas maiores companhias. A Inês foi sempre a nossa companheira de grupo quando os grupos de trabalho eram de três ou mais pessoas. Sem dúvida que a Inês foi uma das pessoas mais inteligentes e mais trabalhadoras que eu já conheci. Sempre que alguém tinha uma dúvida a Inês era a melhor pessoa para a esclarecer, mesmo quando não sabe, a Inês procura e geralmente encontra as respostas. Tenho imenso que lhe agradecer por todo o acompanhamento e ajuda que ela sempre me proporcionou.

Por fim, penso que há ainda um grande conjunto de pessoas a quem eu poderia agradecer mas acho que isso tornaria esta secção menos especial. Este conjunto de pessoas são pessoas que são importantes para mim todos os dias, no entanto não tiveram um impacto tão direto neste trabalho.



## Resumo

O presente relatório diz respeito ao estágio realizado na Norparts, uma das empresas da Create Business que atua no mercado de distribuição de peças de automóveis, tendo as oficinas independentes como clientes alvo. Atualmente a Norparts tem armazéns em Frielas, Porto, Braga e Madrid. As peças são adquiridas a vários fornecedores, são armazenadas nos armazéns, seguem por sua vez para os mecânicos e lojas de venda e por fim chegam aos consumidores finais. Quando chegam aos armazéns, as peças têm que ser arrumadas nas prateleiras correspondentes para o efeito e quando chegam as ordens de encomenda, têm que ser recolhidas no armazém e posteriormente distribuídas pelos clientes. Com base no mapeamento que já estava feito para um dos armazéns, fez-se o mapeamento dos outros armazéns. Neste estágio apresentam-se e implementam-se heurísticas para o problema de determinação de rotas dentro de um armazém (*order picking problem*) e para o problema de determinação de rotas para uma zona de clientes, em que o objetivo é maximizar uma função *profit* garantindo que não se excede um tempo máximo da rota (*orienteeing problem*). Considera-se neste problema a existência de janelas temporais Para a obtenção das rotas de *picking* dentro do armazém utiliza-se a heurística de inserção de menor custo com vista a obter uma primeira solução admissível e posteriormente as heurísticas 2-ótimas e 1-ótimas para o melhoramento da solução obtida anteriormente. Os resultados obtidos usando estas heurísticas foram comparados com os resultados obtidos antes da utilização das heurísticas nos armazéns em que já existiam rotas dentro do armazém. No caso do problema de determinação de rotas numa zona de clientes utiliza-se também uma heurística construtiva baseada no princípio de inserção de menor custo. Descreve-se também a obtenção dos dados para a zona de clientes em estudo e os resultados obtidos pela heurística.

**Palavras Chave:** *orienteeing problem*; *order picking problem*; inserção de menor custo; 2-ótimas; 1-ótimas.





## **Abstract**

This report concerns the internship carried out at Norparts, one of the companies from Create Business that operates in the car parts distribution market with independent workshops as target customers. Norparts currently has warehouses in Frielas, Porto, Braga and Madrid. The parts are purchased from various suppliers, they are stored in warehouses, are distributed to mechanics and stores, and finally reach final consumers. When they arrive at the warehouses, the pieces have to be stored on the corresponding shelves and when the order list arrives, they must be collected at the warehouse and later distributed to customers. Based on the mapping that was already done for one of the warehouses, the other warehouses were mapped. At this internship, heuristics are presented and implemented for the problem of determining routes inside a warehouse (order picking problem) and for the problem of determining routes for a customers' zone, where the objective is to maximize a profit function ensuring that a maximum route time is not exceeded (orienteering problem). In this problem, the existence of time windows is considered. To obtain the picking routes inside the warehouse, the cheapest insertion heuristic is used to obtain a first feasible solution and then the 2-optimal and 1-optimal heuristics to improve the solution obtained previously. The results obtained using these heuristics were compared with the results obtained before using the heuristics in the warehouses where there already existed routes defined. In the case of the problem of determining routes in a customers' area, a constructive heuristic based on the principle of the cheapest insertion is also used. It is also described how to obtain the data for the customers' area under study and the results obtained by the heuristic.

**Key words:** orienteering problem; order picking problem; cheapest insertion; 2-optimal; 1-optimal.



## Índice

Agradecimentos .....	iv
Resumo .....	vi
Abstract .....	viii
Índice de tabelas .....	xii
Índice de figuras .....	xiv
Capítulo 1 .....	1
1. Introdução .....	1
1.1. Revisão bibliográfica relativa ao <i>order picking problem</i> .....	3
1.2. Revisão bibliográfica relativa ao <i>Orienteering Problem</i> .....	4
Capítulo 2 .....	7
2. Determinação de rotas dentro de um armazém .....	7
Capítulo 3 .....	13
3. Determinação das rotas numa zona de clientes .....	13
Capítulo 4 .....	19
4. Apresentação e Análise dos resultados computacionais .....	19
4.1. Apresentação e análise dos resultados relativos ao problema de <i>order picking</i> .....	19
4.2. Apresentação e análise dos resultados relativos ao problema de determinação de rotas numa zona de clientes afeta ao armazém de Frielas .....	27
Capítulo 5 .....	31
5. Conclusões .....	31
Bibliografia .....	33
Anexos .....	35



## **Índice de tabelas**

Tabela 4.1: Testes da ordem das heurísticas .....	20
Tabela 4.2: Testes armazém A de Frielas.....	23
Tabela 4.3: Testes armazém B de Frielas.....	24
Tabela 4.4: Testes armazém A do Porto.....	24
Tabela 4.5: Testes armazém B do Porto.....	25
Tabela 4.6: Testes armazém A de Braga.....	26
Tabela 4.7: Testes armazém B de Braga .....	26
Tabela 4.8: Testes armazém do piso 0 de Madrid .....	27
Tabela 4.9: Testes efetuados para determinar as rotas numa zona de clientes .....	29



## **Índice de figuras**

Figura 2.1: Layout do armazém A de Frielas .....	8
Figura 4.1:Exemplo de uma rota no armazém A de Frielas antes da aplicação das heurísticas .....	21
Figura 4.2:Exemplo depois da aplicação das heurísticas .....	22
Figura 4.3: Exemplo de uma rota de distribuição.....	28





## Capítulo 1

### 1. Introdução

A Create Business é uma empresa que atua no mercado de distribuição de peças de automóveis tendo as oficinas independentes como clientes alvo. Para tal, detém outras empresas que têm contacto direto com os clientes alvo. Uma dessas empresas é a Norparts.

A Norparts tem neste momento armazéns em 4 locais distintos, Frielas, Porto, Braga e Madrid. Cada um dos armazéns funciona ao mesmo tempo de forma individual e em grupo com os outros. De forma individual porque em cada um encontra-se uma equipa preocupada em fazer a gestão do mesmo e em satisfazer os pedidos dos clientes que lhe estão afetos. Por outro lado, em grupo porque apesar de serem armazéns diferentes, muitas vezes é necessário que haja transporte de mercadoria de um destes para outro, para além de que todos pertencem à Norparts e por sua vez à Create Business.

O trabalho diário da Norparts divide-se entre aquilo que acontece no armazém e a parte de distribuição que se passa fora do mesmo, entre outras coisas que não são relevantes para o trabalho em questão. Começamos então por perceber o que se passa dentro do armazém. O armazém está dividido em duas grandes partes. Uma zona é referente aquilo que é designado por peças, e que daqui para a frente designaremos como armazém A, e a outra é composta por pneus, baterias e óleos, e que daqui para a frente designaremos por armazém B. Para simplificar, iremos designar por peças tudo o que não for uma bateria, um pneu ou um óleo. Existe ainda uma terceira zona onde se trata das encomendas. É aqui que se faz a separação das mesmas e onde há a saída da mercadoria para ser mais tarde entregue aos clientes.

Tanto o armazém A como o B, estão divididos em corredores, que por sua vez se dividem em módulos. Cada um destes divide-se em níveis e por fim em cada nível temos a identificação do local de modo que este tenha um carácter único. Tudo o que acabou de ser referido está devidamente identificado. As referências dos locais são como a seguinte #-##-##-##-##, em que o primeiro valor pode ser A ou B consoante o armazém em que se encontra. O segundo e o terceiro caracteres estão reservados para identificar o corredor em que se encontram. O quarto e o quinto caracteres permitem-nos reconhecer o módulo. O sexto e o sétimo caracteres ajudam-nos a identificar o nível, sendo que tipicamente os níveis começam a contar do chão. Por fim, o oitavo e o nono caracteres servem para definir a localização exata.

Ao longo do dia vão chegando pedidos ao sistema. Um pedido pode ter uma ou mais linhas. Cada linha corresponde a um determinado produto, numa mesma linha pode ser pedido uma ou mais quantidades desse produto. Quando há novos pedidos no sistema e estes já têm um tamanho de quinze linhas, independentemente das quantidades, ou quando já se passaram quinze minutos desde a última ordem de separação, cria-se uma nova ordem de separação. Uma ordem de separação consiste em agrupar os pedidos dos clientes. Esta é a etapa onde se iniciam todos os processos pelos quais os produtos irão ter de passar até que estejam prontos para serem entregues aos respetivos clientes.

Após a ordem de separação estar concluída, inicia-se o processo de separação. Este procedimento consiste em ir ao armazém buscar os produtos da ordem de separação por uma determinada ordem. Este processo equivale a determinar uma rota dentro do armazém de modo a recolher todos os produtos da ordem de separação.

O procedimento seguinte consiste em agrupar por cliente os produtos previamente recolhidos. Posteriormente, os produtos seguem para a zona das encomendas prontas a entregar.

Daqui os produtos vão para as carrinhas que farão a sua distribuição pelos respetivos clientes.

Nesta empresa existem muitos problemas onde se podem aplicar técnicas de investigação operacional tornando a gestão mais eficiente. No estágio desenvolvido propôs-se otimizar as rotas dentro dos armazéns de Frielas, Braga, Porto e Madrid e as rotas de distribuição para os clientes de uma determinada zona afeta ao armazém de Frielas.

Em relação às rotas dentro dos armazéns, estas eram inicialmente realizadas em forma de Z. Começam sempre nos últimos corredores e percorre-se todos os corredores em forma de Z. Não estava a ser usado nenhum algoritmo que nos garantisse uma boa solução. Deste modo, foi proposto que se usasse uma heurística para garantir que se obtinha boas rotas de modo a poupar tempo nos percursos realizados. Tanto o armazém A de Frielas, como o armazém B de Frielas, bem como qualquer um dos outros armazéns em qualquer uma das outras localizações está organizado segundo o sistema ABC. Este sistema consiste em classificar cada peça com uma letra do alfabeto. O número de letras usadas para classificar as peças depende de empresa para empresa. As peças são classificadas de acordo com a percentagem de vendas. As mais vendidas são classificadas como A as menos vendidas são classificadas com a última letra que está a ser utilizada. Por exemplo, no caso de uma empresa usar letras entre o A e o F, as peças classificadas com A são as mais vendidas, com B as que se seguem e assim sucessivamente até às peças que são classificadas com F, que representam as peças menos vendidas. Posto isto, as peças que são mais vendidas são colocadas mais perto do local de saída e com um acesso fácil, as segundas peças mais vendidas são colocadas no segundo lugar de mais fácil acesso e mais perto do local de saída e assim sucessivamente até as peças menos vendidas que são colocadas o mais longe possível do local de saída e nos locais de mais difícil acesso.

O segundo desafio proposto neste estágio foi o de otimizar as rotas de distribuição dos produtos aos clientes pertencentes a uma zona servida pelo armazém de Frielas.

Inicialmente as rotas eram decididas pelos entregadores. Quando chegava a hora de realizar as rotas de entrega, caso os entregadores chegassem à conclusão que não tinham tempo de fazer as entregas a todos os clientes que tinham feito um pedido, decidiam quais os clientes que ficariam de fora da rota e qual a ordem de visita dos clientes escolhidos, isto é, a rota a realizar. Estas decisões, embora baseadas na grande experiência dos entregadores eram assim, tomadas sem recurso a métodos científicos que tornassem este processo de decisão mais objetivo e eficiente. Assim, o objetivo será utilizar técnicas heurísticas para fazer a escolha dos clientes a visitar de modo a não exceder um tempo máximo que a rota deve ter e otimizar o percurso que inclui esses clientes.

Este estágio na Norparts consistiu em desenvolver heurísticas para o problema de determinação de rotas dentro de um armazém (*order picking problem*) e para o problema de determinação de rotas para uma zona de clientes em que o objetivo é maximizar uma função *profit* garantindo que não se excede um tempo máximo da rota (*orienteering problem*).

Os problemas de otimização são transversais a várias áreas. O objetivo é determinar uma solução ótima considerando uma determinada medida de performance. Na maioria dos casos essa medida é relativa a custos, os quais se pretendem minimizar, ou relativa a lucros, os quais se pretendem maximizar. Nos problemas reais, normalmente devido à sua grande dimensão e complexidade torna-se difícil obter soluções ótimas num curto espaço de tempo pelo que se recorre à utilização de heurísticas, isto é, métodos de obtenção de soluções admissíveis, idealmente de boa qualidade, num curto espaço de tempo. Nas empresas é fundamental obter boas soluções num curto espaço de tempo.

Apresenta-se de seguida uma sucinta revisão bibliográfica relativa aos dois problemas.

### 1.1. Revisão bibliográfica relativa ao *order picking problema*

Theys *et al.* em [1] resolvem o problema de *order picking* como um caso particular de um *Traveling Salesman Problem* (TSP) clássico. O problema inicial é reformulado como um TSP calculando a distância do caminho mais curto entre cada dois pontos de *picking* (incluindo o depósito), usando distâncias de Manhattan. A distância de Manhattan entre dois pontos com coordenadas  $(x_1, y_1)$  e  $(x_2, y_2)$  é  $|x_1 - x_2| + |y_1 - y_2|$ . Contudo, se dois pontos de *picking* estiverem em partes opostas no mesmo bloco, a distância de Manhattan não pode ser usada porque o caminho mais curto tem de passar por um dos dois corredores adjacentes ao bloco. Assim, estes autores calculam o tamanho dos dois possíveis caminhos e mantém o mais curto. Usando este processo para calcular as distâncias entre os vários pontos de *picking* (incluindo o depósito), obtêm uma matriz de distâncias simétrica que será o *input* para uma heurística de TSP. Após a obtenção dessa matriz de distâncias, usam qualquer heurística existente atualmente para um problema clássico de TSP de modo a obter uma redução da distância da rota.

Daniels *et al.* em [2] dada a complexidade de obter até um limite inferior para o problema de *order picking*, focaram-se primeiro numa heurística que lhes permitisse contruir uma solução para o problema. As heurísticas para o TSP do vizinho mais próximo standard e do arco mais curto foram primeiramente modificadas para irem ao encontro das definições do novo problema. Tendo em conta que a performance destas técnicas deve ser mais pobre na aproximação de soluções ótimas para o problema de *order picking* do que para o TSP, usaram-se métodos de amostragem de áreas promissoras da região admissível, incluindo versões aleatórias das heurísticas do vizinho mais próximo modificado e do arco mais curto, assim como abordagens de procura tabu para o problema de *order picking*.

Pansart *et al.* em [3] reduzem em primeiro lugar o número de pontos de *picking* e seguidamente reduzem o número de arcos do grafo, de modo a diminuir substancialmente a dimensão do problema, de modo a que seja possível encontrar a solução ótima. Para reduzir o número de vértices (que representa o número de pontos de *picking*) são usados dois algoritmos. O primeiro baseia-se em verificar se existem pontos de *picking* em corredores, no caso de existirem mais do que dois pontos de *picking* num corredor, eliminam-se todos os pontos de *picking*, exceto os dois extremos e acrescenta-se um arco entre estes. Deste modo, todos esses pontos de *picking* são representados por esse arco e este tem de fazer parte da rota final. O segundo algoritmo é usado por exemplo quando não é possível adicionar restrições. Nestes casos usa-se a matriz de distâncias. É possível remover vértices desde que se mantenha a maior diferença entre vértices de um corredor que estejam na rota e vértices de outro corredor que ainda não estejam na rota.

Chabot *et al.* em [4] sugerem e comparam três diferentes abordagens para a resolução do problema. A primeira é o método usado pelo seu parceiro industrial. Para cada rota de *picking*, começa-se com o produto que está na secção mais longe e no nível mais elevado. De seguida, adiciona-se o próximo produto possível que esteja no próximo nível mais elevado e na mesma secção. Se não é possível ir buscar mais nenhum produto na mesma secção, segue-se para a próxima secção mais longe onde é possível apanhar um produto. Desta maneira, escolhe-se sempre o produto na secção mais longe e no nível mais elevado, que respeite as restrições de movimento de segurança. A segunda abordagem consiste em implementar uma *adaptive large neighborhood search* (ALNS). Esta é composta por um conjunto de heurísticas simples de destruição e reconstrução, de modo a encontrar uma solução melhor em cada iteração. Pode considerar-se uma solução inicial para acelerar o processo de busca e a convergência do algoritmo. Implementou-se uma heurística de inserção sequencial rápida, que executa uma heurística *greedy* para a melhor inserção de um produto de cada vez. A ALNS seleciona um de muitos operadores de destruição e reconstrução para cada iteração. Cada operador é selecionado com uma probabilidade que depende da sua performance passada e é usado um critério de aceitação de

*simulated annealing*. Na última abordagem é utilizado um algoritmo clássico de *branch-and-cut* em que são usadas desigualdades lineares válidas como planos de corte para fortalecer uma relaxação linear em cada nodo da árvore de *branch-and-bound*. As restrições são primeiramente relaxadas e só depois adicionadas à árvore de *branch-and-bound* se se descobrir que estas são violadas. O algoritmo apresentado é um esquema de *branch-and-cut* em que são geradas restrições de desigualdades de capacidade arredondada e são adicionadas ao modelo quando se descobrir que estas são violadas. Num nodo genérico da árvore de procura, um programa linear que contém o modelo com um subconjunto de restrições de eliminação de subcircuitos e restrições de integralidade relaxadas resolvidas, realiza-se uma procura por desigualdades violadas, e algumas destas são adicionadas ao programa que é depois reotimizado. Este processo repete-se até que uma solução admissível ou dominada seja alcançada, ou até que não haja mais cortes a adicionar. Neste ponto, ocorre *branching* de uma variável fracionária.

Em [5] os mesmos autores implementam tal como para o problema anterior uma heurística *adaptive large neighborhood search* (ALNS) e um algoritmo do tipo *branch-and-cut* para um problema de *order picking* incorporando várias características especiais dos produtos que aparecem em aplicações reais tais como o peso e a fragilidade dos produtos.

## 1.2. Revisão bibliográfica relativa ao *Orienteering Problem*

Archetti *et al.* em [6] sugerem três diferentes algoritmos, no entanto todos seguem o mesmo esquema geral. Dada uma solução qualquer  $s$ , troca-se para uma solução  $s'$ . Aplica-se de seguida uma procura tabu em  $s'$  com vista a tentar melhor a solução. A solução resultante  $s''$  é comparada com  $s$ . Após este processo sugerem dois procedimentos diferentes, uma estratégia VNS ou uma estratégia generalizada da procura tabu. No primeiro procedimento referido,  $s''$  torna-se a nova solução apenas se for melhor que  $s$ . No segundo procedimento referido,  $s''$  torna-se a nova solução mesmo que seja pior que  $s$ . O próximo passo é o de tornar a solução obtida numa solução admissível. Os autores obtêm este objetivo reduzindo sucessivamente a inadmissibilidade até que a solução se torne admissível. Se esta já for admissível à partida então o primeiro *output* é a mesma solução do *input*. Após a obtenção de uma solução admissível, realiza-se uma procura de outras soluções numa determinada vizinhança com vista a obter uma melhor solução. Para tentar determinar uma solução melhor ainda, usa-se uma procura tabu. Ao longo de todo o percurso, cada vez que se obtém uma solução não admissível, usa-se o algoritmo de obtenção de uma solução admissível.

Ramesh *et al.* em [7] afirmam que as heurísticas existentes para o problema *orienteering* usam funções de custo baseadas na distância euclidiana e desenvolvem uma heurística que, segundo os autores, é eficiente para qualquer função custo. Comparam os resultados computacionais obtidos com o algoritmo desenvolvido com os obtidos por outros métodos existentes e concluem que o algoritmo que apresentam é eficiente para resolver problemas práticos com funções custo arbitrárias. O algoritmo apresentado para o GOP (*Generalized Orienteering Problem*) consiste em 4 fases: inserção de vértices; melhoramento do custo; eliminação de vértices e inserção maximal. Os autores começam por usar um determinado procedimento de inserção em que vão inserindo vértices até que ao fazerem uma determinada inserção, o custo do caminho ultrapasse um determinado limite, quando isto acontecer, o procedimento termina e o caminho fica definido como estava antes de este último vértice ter sido inserido. Após a obtenção de um caminho, utiliza-se a heurística 2-optimal. Caso a razão entre o custo deste novo caminho e o custo do primeiro caminho obtido seja inferior a um determinado valor, realiza-se a heurística 3-optimal. Neste ponto verifica-se a admissibilidade do caminho. Se o caminho for admissível segue-se com o procedimento de inserção de vértices maximal partindo da solução obtida anteriormente e pára-se antes de se obter uma solução não admissível. Se o caminho não for admissível existem duas hipóteses, caso o custo do caminho obtido após a utilização das heurísticas 2-optimal e 3-optimal seja superior a um

determinado valor, então seleciona-se um vértice para sair do caminho. Se ao retirar qualquer vértice do caminho, nenhum resultar num melhoramento do custo do caminho, segue-se com o procedimento de inserção maximal. Se ao retirar um vértice do caminho, houver pelo menos um que resulte num melhoramento do custo do caminho, então apaga-se o vértice desse caminho e volta para o primeiro procedimento de inserção. No caso de o valor obtido após a utilização das heurísticas não ser superior ao valor referido anteriormente, então volta-se ao procedimento de inserção dos vértices.

Vansteenwegen *et al.* em [8] focam-se no problema de *orienteering* (orientação). Em várias secções do artigo, definem e discutem o problema de orientação, o problema de orientação em equipas, o problema de orientação com janelas temporais e o problema de orientação em equipas com janelas temporais. As soluções apresentadas neste artigo para cada um dos problemas referidos anteriormente, são as soluções que se encontram na literatura.

Archetti *et al.* em [9] abordam dois grandes problemas, o CTOP (*capacited team orienteering problem*) e o CPTP (*capacitated profitable tour problem*). Para resolver os problemas, os autores propõem algoritmos diferentes para cada, no entanto, ambos se apoiam no esquema semelhante ao do *branch-and-price*. Implementaram três meta heurísticas, um algoritmo de pesquisa local e dois algoritmos de procura tabu. A estrutura principal dos algoritmos é a mesma que a apresentada em [6] que foi por sua vez adaptada para lidar com as restrições de capacidade e com os diferentes objetivos dos dois problemas apresentados.

Toth *et al.* em [10] apresentam modelos, variantes e métodos de resolução para o problema de orientação.



## Capítulo 2

### 2. Determinação de rotas dentro de um armazém

Neste capítulo descreve-se o problema de determinação de rotas dentro de um armazém e apresentam-se as heurísticas usadas.

Como foi dito anteriormente os armazéns estão divididos em corredores, que por sua vez se dividem em módulos, cada um destes divide-se em níveis e por fim em cada nível temos a identificação do local de modo que este tenha um carácter único.

Consideremos por exemplo a referência A03050412. Esta referência indica-nos que o produto pretendido está no armazém A, corredor três, módulo cinco, nível quatro e posição doze.

Para calcular a distância entre dois quaisquer pontos do armazém não podemos ignorar a geometria do mesmo. Uma vez que o armazém se encontra dividido em corredores pelas estantes, se pensarmos em dois produtos que estão no lado oposto da mesma estante (corredores diferentes), para calcularmos a distância entre estes dois produtos, tendo em conta que não podemos atravessar a estante, teríamos de nos deslocar até ao fim do corredor para mudar para o corredor onde se encontra o outro produto. Entre qualquer par de pontos do mapeamento que sejam adjacentes, considera-se uma distância de um. Através de todas as adjacências calcula-se o caminho mais curto entre cada par de pontos do mapeamento.

Antes da aplicação de qualquer heurística é sempre necessário fazer a correspondência entre as referências dos armazéns e a maneira como as vamos representar na codificação. Este processo já estava feito para o armazém A de Frielas. A forma de representação escolhida foram células do Excel. Cada célula é tipicamente um ponto e corresponde aproximadamente a um metro no armazém, isto para tentar fazer uma representação o mais real possível dos mesmos.

A imagem seguinte representa o mapeamento do armazém de Frielas. A última linha, que se encontra a negrito, representa a identificação do corredor. O corredor zero e onze, não estão de momento a ser utilizados, no entanto o mapeamento está pronto para considerar o serviço dos mesmos. Olhando agora apenas para o corredor um, por exemplo, entra-se pelo ponto 45, e à esquerda temos as estantes numeradas com os números ímpares e à direita as estantes numeradas com os números pares. Ou seja, a primeira estante à esquerda é a número um, a primeira à direita é a número dois, e assim sucessivamente. Ao acabarmos esse conjunto de estantes a numeração continua no conjunto seguinte, até às últimas duas estantes desse corredor, que no caso deste armazém são a vinte e cinco e a vinte e seis. Olhando agora por exemplo para a estante nove do corredor um. Dentro desta estante existem várias prateleiras, cada prateleira geralmente representa um nível, sendo que em certos casos pode representar mais do que um nível. O nível mais perto do chão é o um e à medida que vamos subindo, o número do nível vai aumentando. Este armazém possui uma particularidade, algumas peças situam-se no topo das estantes, para não se confundir com os outros níveis, estas foram definidas com o nível zero. Para facilitar a identificação destas localizações, foram todas definidas pelos módulos pares, ou seja, os finais das estantes em cada corredor encontram-se nos módulos seis, dez, dezasseis e vinte e seis. Dentro de cada nível existe um determinado número de localizações, cada localização é definida por um último número que representa a sua posição na estante. O ponto zero é o ponto de início e de fim da rota. É o ponto central mais perto de onde normalmente quem irá fazer a rota de *picking* irá começar.

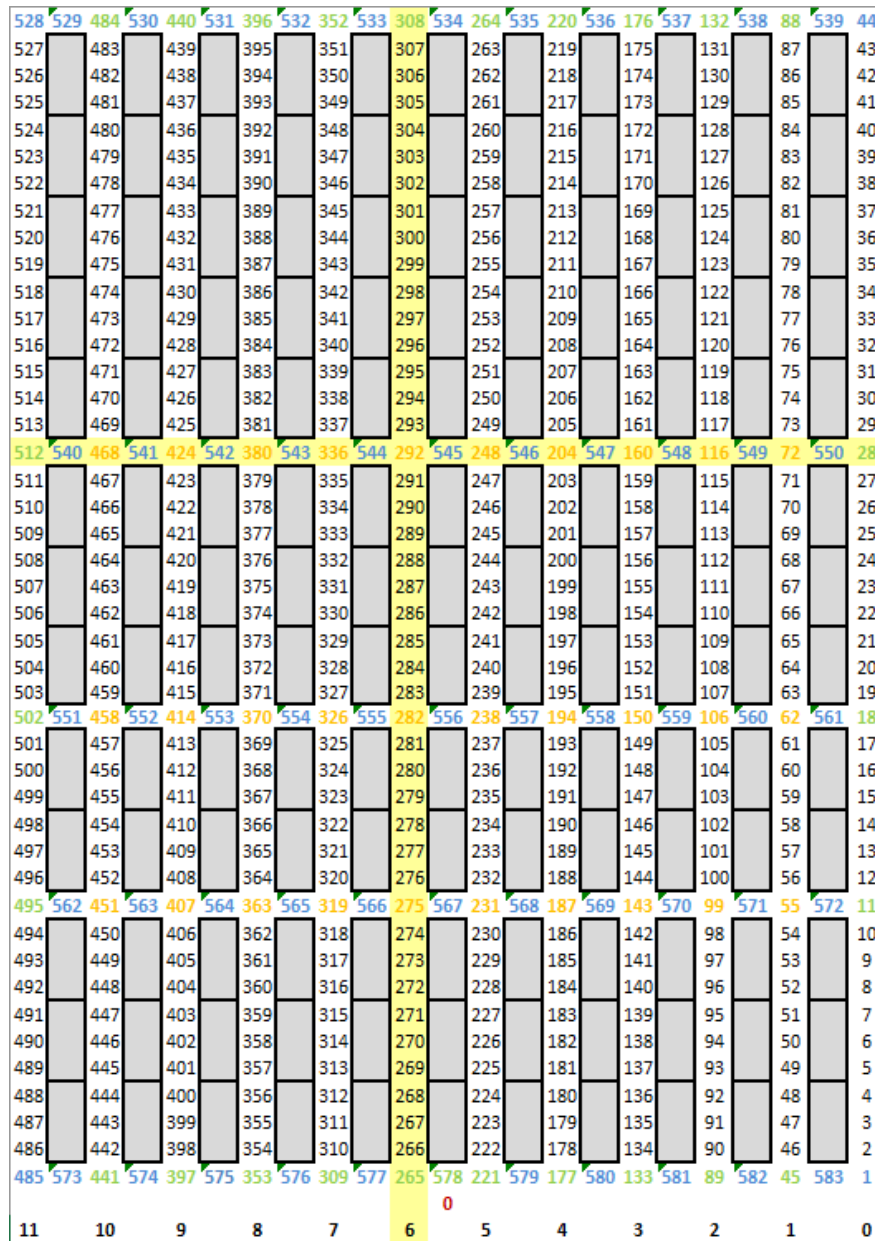


Figura 2.1: Layout do armazém A de Frielas

Consideremos o seguinte exemplo correspondente a uma ordem:

*picking* [1] = "A10100010";

*picking* [2] = "A07030615";

*picking* [3] = "A05010414";

*picking* [4] = "A04020403";

*picking* [5] = "A01090603";

Esta ordem tem cinco pontos de *picking*. O primeiro passo é fazer a correspondência entre cada um destes pontos e todos os pontos mapeados. À exceção dos níveis zero, o resto dos níveis são apenas identificados pelo corredor e pelo módulo em que se encontram. A primeira localização encontra-se no corredor dez, na décima estante, nível zero e posição dez. Como o nível deste ponto é o zero, o ponto



do mapeamento que lhe corresponde é o ponto 552. As restantes localizações têm nível diferente de zero, por isso só precisamos de ter em conta o corredor e o módulo. A segunda localização encontra-se no módulo três do corredor sete. O ponto que lhe corresponde é o ponto 314. A terceira localização encontra-se no módulo um do corredor cinco. O ponto que lhe corresponde é o ponto 223. A quarta localização encontra-se no módulo dois do corredor quatro. O ponto que lhe corresponde é o ponto 179. A última localização encontra-se no módulo nove do corredor um. O ponto que lhe corresponde é o ponto 60. No código correspondente ao mapeamento do armazém, este processo realizou-se para todas as localizações do armazém, de modo que quando se pretender calcular a rota é necessário fazer apenas a correspondência entre as localizações recebidas e o ponto do armazém que lhe corresponde e que já está definido.

Com base no que já estava feito, adaptou-se a ideia para os armazéns A e B do Porto e de Braga, o armazém B de Frielas e o piso zero do armazém de Madrid.

Neste momento cada estante é representada por apenas um ponto. Esta representação faz com que referências a uma distância de aproximadamente dois metros e setenta sejam enviadas para o mesmo ponto de forma indiferenciada. Futuramente, para um melhor ajustamento à realidade, pretende-se representar cada estante por três pontos.

Para a aplicação das heurísticas foi necessário saber quais as distâncias entre os pontos do armazém que representam pontos de *picking*, para isto usou-se o Algoritmo de Floyd-Marshall [11], que permite obter o valor dos caminhos mais curtos entre todos os pontos. Este algoritmo já estava implementado para o caso do armazém A de Frielas. Para os outros armazéns foi necessário fazer o mapeamento dos armazéns, determinar as adjacências entre os vértices e aplicar o algoritmo de Floyd-Marshall.

Para a obtenção das rotas de *picking* dentro do armazém utilizou-se a heurística de inserção de menor custo [12] com vista a obter uma primeira solução admissível e posteriormente a heurística 2-optimal [13] para o melhoramento da solução obtida anteriormente.

A heurística construtiva já estava implementada para o armazém A de Frielas. A heurística utilizada foi o algoritmo *cheapest insertion*. Esta heurística é uma heurística *greedy*, ou seja, um algoritmo que toma sempre a melhor opção disponível em cada momento, mas sem considerar os cenários que se seguirão. Neste tipo de heurística toma-se uma decisão que é a melhor naquela iteração, mas que condiciona as decisões tomadas posteriormente.

A heurística da inserção de menor custo (*cheapest insertion*) consiste em inserir numa rota existente o vértice que ainda não pertence à rota e a que corresponde o menor custo de inserção na rota. Inicia-se a rota com dois vértices, tal que a distância entre os mesmos corresponde à menor distância entre qualquer par de vértices.

No problema de determinação de rotas dentro dos armazéns assumimos que a matriz das distâncias é simétrica.

Apresenta-se seguidamente o algoritmo correspondente à heurística de inserção de menor custo.

#### Algoritmo 2.1. Heurística de Inserção de menor custo

Input: matriz com as distâncias entre qualquer par de vértices  $[d_{ij}] \ i, j \in V = \{\text{vértices}\}$

Determinar os vértices a que corresponde a aresta de menor custo, isto é, os vértices  $i^*$  e  $j^*$  tal que  $d_{i^*j^*} = \min_{i,j \in V} d_{ij}$

Seja  $S$ , a solução corrente. Neste caso tem-se  $S = \{(i^*, j^*), (j^*, i^*)\}$  e  $V_S = \{i^*, j^*\}$ .  $Custo = 2 \times d_{i^*j^*}$

Enquanto  $(V \setminus V_S \neq \emptyset)$  fazer:

Calcular para cada  $j \in V \setminus V_S$  e para cada  $(k, l) \in S$   
$$\min_{(k,l) \in S} d_{kj} + d_{jl} - d_{kl}$$

Seja  $j^* \in V \setminus V_S$  o vértice a que corresponde o mínimo e  $(k^*, l^*) \in S$  a correspondente aresta

$V_S = V_S \cup \{j^*\}$   
 $S = S \cup \{(k^*, j^*), (j^*, l^*)\}$   
 $Custo = Custo + d_{k^*j^*} + d_{j^*l^*} - d_{k^*l^*}$

Output:  $S$  e  $Custo$

Depois de obter uma solução admissível com recurso a uma heurística construtiva utilizou-se uma heurística de melhoramento local com vista à obtenção de soluções admissíveis com melhor valor. Tal procedimento faz sentido uma vez que a melhor decisão tomada numa altura na heurística construtiva *greedy* pode não ser a melhor a nível global.

Com o objetivo de melhorar a solução obtida na fase construtiva utilizou-se numa primeira fase uma heurística 2-optimal, e sobre a solução obtida com esta heurística de melhoramento local aplicou-se uma espécie de heurística 1-optimal para vértices.

A heurística 2-optimal consiste em, dada uma solução admissível obtida na fase construtiva, retirar a essa solução duas arestas que não sejam consecutivas e reconstruir a solução adicionando duas novas arestas de modo a manter um circuito hamiltoniano. Uma vez que o problema em questão contém distâncias simétricas, existe apenas uma maneira de adicionar duas novas arestas. Depois repete-se o processo para todos os conjuntos de duas arestas não adjacentes. Após a obtenção de um novo circuito, calcula-se segundo um critério da empresa, qual a direção da rota. Define-se a vizinhança 2-optimal de uma solução admissível  $T$  para o problema do caixeiro viajante como:

$N_2(T) = \{\text{percursos admissíveis para o TSP que diferem de } T \text{ no máximo em 2 arestas}\}.$

Apresenta-se a seguir, resumidamente, o algoritmo corresponde à heurística 2-optimal.

#### Algoritmo 2.2. Heurística 2-optimal

Input: Percurso admissível  $T$  para o problema

Passo 1. Obter os percursos  $P \in N_2(T)$

Passo 2. Escolher o percurso  $P \in N_2(T)$  a que corresponde o custo mínimo

Passo 3. Se  $Custo(P) < custo(T)$  Fazer  $T \leftarrow P$  e voltar ao Passo 1

Output:  $T$  e Custo

A heurística 1-optimal sugerida consiste em retirar um nodo do circuito, fechar o circuito e depois verificar qual o custo ao inserir o nodo que foi retirado entre todos os dois nodos adjacentes do circuito e depois inserir no local onde obtivermos um melhor valor da função objetivo. Seguidamente repete-se o processo para todos os nodos.

Apresenta-se de seguida o algoritmo corresponde à heurística 1-optimal usada.

#### Algoritmo 2.3. Heurística 1-optimal

Input: Percurso admissível  $T$  para o problema

Passo 1. Retirar um nodo qualquer do percurso  $T$

Passo 2. Fechar o circuito  $T$

Passo 3. Obter todos os circuitos possíveis ao inserir o nodo retirado no passo 1, entre todos os pares de nodos adjacentes do circuito

Passo 4. Inserir o nodo retirado no passo 1 no local do circuito em que  $Custo(T)$  é mínimo

Passo 5. Repetir os passos de 1 a 4 para todos os nodos do circuito

Output:  $T$  e Custo



## Capítulo 3

### 3. Determinação das rotas numa zona de clientes

Neste capítulo apresenta-se o problema de determinação das rotas para servir os clientes pertencentes a uma determinada zona. Os clientes servidos pela empresa estão divididos em várias zonas. Cada zona de clientes é servida por um veículo com capacidade suficiente para servir todos os clientes. Neste problema a restrição corresponde ao tempo máximo da rota, o que pode levar a que nem todos os clientes possam ser servidos nessa rota, tendo que ser servidos posteriormente noutra rota que se realizará mais tarde, isto é, com atraso relativamente ao prazo previsto para receber a encomenda.

As rotas são realizadas por zona e com horários fixos. Por exemplo, os carros podem sair às 9h e depois sair novamente às 11h. Neste exemplo, os carros que saem às 9h têm um tempo máximo para realizar a rota de 2h, uma vez que às 11h têm de estar de volta para poder começar uma nova rota. Posto isto, ao elaborar cada rota (qualquer hora, qualquer zona) é necessário ter em atenção o tempo máximo para a realização da mesma, bem como os horários de atendimento dos clientes que queremos servir.

A heurística de inserção desenvolvida neste trabalho para o problema de determinação de rotas numa zona de clientes baseia-se numa heurística apresentada por Gendreau *et al.* em [9]. Estes autores propõem uma generalização de uma heurística de inserção para o problema do caixeiro viajante com janelas temporais. Trata-se de uma generalização da heurística apresentada por Gendreau *et al.* em [10]. Estes autores começam por inserir todos os vértices seguindo uma ordem não decrescente do tamanho das respetivas janelas temporais. Construindo assim um primeiro caminho. Após a construção do caminho, verifica-se se este é admissível. Se assim for aplica-se um procedimento de pós-otimização que consiste basicamente em remover um vértice da solução e reinseri-lo. Começando com o sucessor do primeiro vértice que representa o depósito, remove-se à vez cada vértice do caminho e reinsere-se esse mesmo vértice no caminho usando o procedimento GENI. Se a solução for melhor que a anterior então é aceite. Caso contrário, em particular se esta solução for não admissível, é descartada. Se ao aplicar este procedimento não ocorrer melhoramento no valor da função objetivo, pára-se, se tal não se verificar, repete-se o procedimento de remover e reinserir os vértices tal como descrito acima, começando novamente no sucessor do primeiro vértice que é o depósito. Se após construir o primeiro caminho se verificar que é admissível, tenta-se introduzir os vértices que não pertencem ao caminho entre dois dos seus vizinhos mais próximos, usando o procedimento GENI. Se não for encontrada nenhuma inserção admissível, segue-se para outro vértice que ainda não se tenha tentado inserir e assim sucessivamente até serem tentados todos os vértices que não pertencem ao caminho. Se se encontrar pelo menos uma inserção admissível para um dos vértices, escolhe-se a melhor e atualiza-se o caminho. Posto isto, verifica-se novamente a admissibilidade do caminho e repete-se um dos processos descritos acima, dependendo se o caminho é ou não admissível. Caso não se encontre nenhuma solução admissível após se ter tentado inserir todos os vértices, remove-se do caminho o vértice sucessor do primeiro e coloca-se no último lugar do grupo dos vértices que não pertencem ao caminho. Limita-se o número de vezes que se repete esse processo de remover o vértice para que não se entre num ciclo infinito. Caso este limite tenha sido atingido, pára-se e conclui-se que não se conseguiu encontrar uma solução admissível. Caso o limite não tenha sido ainda atingido, repete-se o processo de tentar inserir os vértices que não se encontram no caminho.

Desenvolveu-se para o problema de determinação de rotas numa zona de clientes uma heurística construtiva. Definem-se em primeiro lugar alguns parâmetros e variáveis usados no algoritmo correspondente à heurística usada.

**Definição de parâmetros e variáveis:**

$X$  – Conjunto de todos os clientes que se quer servir

$Z$  – Conjunto de clientes que ainda falta inserir na rota

$R$  – Conjunto de clientes na rota atual por ordem de visita

$T$  – Tempo total da rota a ser realizada, em minutos

$T_{\max}$  – Tempo máximo admissível para percorrer a rota, em minutos

$e_i$  – Momento de abertura da loja do cliente  $i$ . Limite inferior da janela temporal do cliente  $i$  na parte da manhã,  $i \in X$ , em minutos

$f_i$  – Momento de fecho da loja do cliente  $i$ . Limite superior da janela temporal do cliente  $i$  na parte da manhã,  $i \in X$ , em minutos

$g_i$  – Momento de abertura da loja do cliente  $i$ . Limite inferior da janela temporal do cliente  $i$  na parte da tarde,  $i \in X$ , em minutos

$h_i$  – Momento de fecho da loja do cliente  $i$ . Limite superior da janela temporal do cliente  $i$  na parte da tarde,  $i \in X$ , em minutos

$c_{ij}$  – Tempo necessário para viajar do cliente  $i$  ao cliente  $j$ ,  $i, j \in X$ , em minutos

$t_s$  – tempo de serviço, valor fixo e igual para todos os clientes, em minutos

Proximidade entre as janelas temporais, da manhã, dos clientes  $i$  e  $j$ , em minutos:

$$r_{ij}^1 = \min\{f_j, f_i + c_{ij}\} - \max\{e_j, e_i + c_{ij}\}, i, j \in X$$

Proximidade entre as janelas temporais, da tarde, dos clientes  $i$  e  $j$ , em minutos:

$$r_{ij}^2 = \min\{h_j, h_i + c_{ij}\} - \max\{g_j, g_i + c_{ij}\}, i, j \in X$$

Sendo  $p = p_1 + p_2$ , onde  $p_1 = p_2 = p/2$ , definem-se os seguintes conjuntos para cada cliente  $i$ ,  $i \in Z$ .

$N_{p_1}^+(i)$  –  $p_1$  clientes com menor  $c_{ij}$  (candidatos a serem sucessores de  $i$ ),  $i \in Z, j \in R$

$N_{p_2}^+(i)$  –  $p_2$  clientes com maior  $r_{ij}$  (candidatos a serem sucessores de  $i$ ),  $i \in Z, j \in R$

$N_{p_1}^-(i)$  –  $p_1$  clientes com menor  $c_{ji}$  (candidatos a serem antecessores de  $i$ ),  $i \in Z, j \in R$

$N_{p_2}^-(i)$  –  $p_2$  clientes com maior  $r_{ji}$  (candidatos a serem antecessores de  $i$ ),  $i \in Z, j \in R$

Estes conjuntos foram considerados com vista a reduzir o número de troços que iríamos estudar aquando da inserção de cada um dos clientes.

Ao considerarmos estes conjuntos para a inserção de um dado cliente  $i$ , chegou-se à conclusão que estaríamos a estudar um total de  $2p$  troços onde poderíamos inserir o cliente  $i$ .

Tendo em conta que uma rota formada por  $2p$  clientes contém  $2p+1$  troços, até estarem inseridos  $2p$  clientes, o número de troços a estudar, será sempre o número total de troços da rota. Logo, é menos

dispendioso, em termos de unidades de CPU, em segundos, estudar diretamente todos os troços do que estudar primeiro os conjuntos, para depois chegar à conclusão que iremos estudar todos os troços na mesma.

Posto isto, os primeiros  $2p$  clientes são inseridos estudando todas as possibilidades de inserção como será descrito mais à frente. Considerou-se neste trabalho para  $p$  o valor 2.

Sejam:

$s_i$  – segmento a que pertence o cliente  $i$ ,  $i \in X$ , (se o cliente  $i$  pertence ao segmento A,  $s_i = 100$ , se o cliente  $i$  pertence ao segmento B,  $s_i = 50$ , se o cliente  $i$  pertence ao segmento C,  $s_i = 10$ )

$v_i$  – valor da encomenda do cliente  $i$ ,  $i \in X$ , em euros

$D_i$  – tempo decorrido desde que o pedido mais antigo do cliente  $i$  foi colocado,  $i \in X$ , em minutos

$x_i = \begin{cases} 1 & \text{se o cliente } i \text{ é incluído na rota} \\ 0 & \text{c. c.} \end{cases}, i \in X$

$n$  – Número de clientes que queremos servir, cardinalidade de  $X$

$\beta_1$  – peso dado ao segmento do cliente  $i$ , valor entre 0 e 1

$\beta_2$  – peso dado ao valor da encomenda do cliente  $i$ , valor entre 0 e 1

$\beta_3$  – peso dado ao tempo decorrido desde que o pedido mais antigo do cliente  $i$  foi colocado, valor entre 0 e 1

Para inserir o cliente  $i \in Z$  entre os clientes  $j, k \in R$  calcula-se:

$$v_{ijk} = \frac{c_{ji} + c_{ik} - c_{jk}}{\beta_1 \frac{\sum s_m x_m}{\sum s_m} + \beta_2 \frac{\sum v_m x_m}{\sum v_m} + \beta_3 \frac{\sum D_m x_m}{\sum D_m}}, i \in Z, j, k \in R, m \in X$$

O critério de inserção usado nesta heurística baseia-se no valor de  $v_{ijk}$  para os clientes  $i \in Z, j, k \in R$ .

O denominador do critério de inserção é a função objetivo do problema de determinação de rotas em estudo. Nesta função objetivo, são considerados três parâmetros. O primeiro parâmetro diz respeito ao segmento a que pertence cada cliente. A classificação dos clientes em segmentos obedece a diversos fatores tais como a antiguidade do cliente, o volume das suas encomendas, a frequência das suas encomendas, etc. A cada segmento de clientes corresponde uma prioridade dada pelo valor de  $s_i$ . Os clientes do segmento A são os clientes com maior prioridade seguindo-se por ordem decrescente de prioridade os clientes dos segmentos B e C. O segundo parâmetro é o valor da encomenda do cliente, encomendas com maior valor monetário têm maior prioridade. O terceiro e último parâmetro é o tempo decorrido desde a primeira encomenda, um cliente que esteja à espera há mais tempo tem maior prioridade.

O objetivo neste problema é satisfazer o maior número de clientes, tendo em conta os três fatores acima referidos. Para tal, criou-se o critério de inserção acima descrito, que tem em conta para além dos três fatores, também a distância percorrida, uma vez que, quanto menor for o tempo perdido entre viagens, maior é o número de clientes que se pode servir numa rota.

Criam-se as matrizes CM e CT, que contêm os valores dos  $r_{ij}^1$ ,  $r_{ij}^2$ , respetivamente.

### **Algoritmo**

#### *Input*

N: número de clientes a servir

Matrizes C, CM, CT e CS

$s, D, v, \beta_1, \beta_2, \beta_3, p1, p2, p$

$p=2$  e  $p1=p2=1$

Inicialização

$X = \{1, \dots, n\}$

$Z = X$

$R = \{ \}$

$T = 0$

Começar com uma rota inicial composta só pelo armazém (cliente 0)

Adicionar a esta rota inicial (2p) clientes do seguinte modo:

Calcular o valor de  $v_{ijk}$  para todos os clientes  $i$  e considerando todos os possíveis troços de inserção que respeitem as condições de inserção.

Condições de inserção do cliente  $i$  entre os clientes  $j$  e  $k$ :

C1: O valor dos dois  $r_{ij}$  é maior ou igual a 0, uma vez que um  $r_{ij} = 0$ , significaria uma de duas, ou não seria possível chegar ao cliente  $j$ , saindo do cliente  $i$  no momento de abertura e demorando o menor tempo possível entre o cliente  $i$  e o cliente  $j$ , ou chegaríamos ao cliente  $j$  antes do mesmo abrir. Ou seja, ter qualquer  $r_{ij}$  menor que 0 significaria incompatibilidade em servir o cliente  $i$  e de seguida o cliente  $j$ .

C2: O tempo necessário para percorrer a rota não ultrapasse  $T_{\max}$ .

C3: A janela temporal do cliente  $i$  é respeitada.

C4: As janelas temporais de todos os clientes que pertencem à rota e que sucedem o cliente  $i$  são respeitadas.

Insere-se na rota o cliente  $i$  que tiver melhor valor de  $v_{ijk}$  no troço onde esse valor for obtido.

Atualizar  $R$ , inserindo  $i$  entre  $j$  e  $k$ . Atualizar  $Z$ , removendo  $i$ . Atualizar  $T$ , acrescentando o valor de  $t_s$  ao valor de  $T$ .



**Processo de inserção (em cada momento de inserção, iremos tentar inserir na rota atual o cliente  $i \in Z$ ):**

**Repetir:**

Calcular para todos os clientes  $i \in Z$ , os quatro conjuntos definidos anteriormente ( $N_{p1}^+$ ,  $N_{p2}^+$ ,  $N_{p1}^-$  e  $N_{p2}^-$ ).

Se ao calcular os conjuntos para um certo cliente  $i$ , um dado cliente  $j$  for escolhido para um dos conjuntos, por exemplo  $N_{p1}^+$ , este é excluído no cálculo do conjunto correspondente ( $N_{p2}^+$ ), uma vez que ser incluído no primeiro conjunto já significa ser um possível sucessor de  $i$ . O mesmo acontece entre  $N_{p1}^-$  e  $N_{p2}^-$ .

Considerar de entre todos os clientes pertencentes a algum dos conjuntos de  $i$  apenas os que verificam as condições de inserção C1 a C4.

Calcular o valor de  $v_{ijk}$  para cada um dos troços encontrados através dos conjuntos que respeite as condições indicadas acima.

De entre os valores de  $v_{ijk}$  calculados, inserir na rota o cliente  $i$  no troço  $jk$  que apresentar menor valor de  $v_{ijk}$ .

Ao inserir um cliente, aumentar  $T$  para  $T + t_s$ . Inserir  $i$  entre  $j$  e  $k$  em  $R$ . Remover  $i$  de  $Z$ .

**Até que se verifique uma das seguintes condições:**

- O tempo da rota seja igual a  $T_{max}$ ;
- Não seja possível inserir mais nenhum cliente na rota (seja porque nenhum dos clientes que não pertence à rota não verifica as condições de inserção C1 a C4, seja por já termos inserido todos os clientes de  $Z$ ).



## Capítulo 4

### 4. Apresentação e Análise dos resultados computacionais

Neste capítulo apresenta-se a descrição dos ficheiros necessários para a obtenção das rotas dentro dos armazéns e apresentam-se os resultados obtidos para os armazéns de Frielas, Braga, Porto e Madrid. Comparam-se os resultados obtidos com as rotas efetuadas antes desta otimização. Descreve-se também a obtenção dos dados para a zona de clientes em estudo e os resultados obtidos pela heurística.

#### 4.1. Apresentação e análise dos resultados relativos ao problema de *order picking*

Para determinar as rotas dentro do armazém é necessário construir um ficheiro de *input*. Esse ficheiro deve conter a geografia do armazém, o número total de pontos de *picking* e para cada ponto de *picking* a referência que lhe corresponde. No anexo A podemos ver um exemplo no qual a geografia é representada pelo número 1010 que é a geografia do armazém de Frielas, o número total de pontos de *picking* é 7 e cada uma das linhas que se segue representa uma referência. Este ficheiro é construído com base numa ordem de separação.

Como ficheiros de *output*, teremos um ficheiro que contém apenas o valor da solução e o tempo de CPU, em segundos, para obter a solução, como se pode ver no anexo B.

O segundo ficheiro de *output* contém o número de pontos da rota, o custo da rota, o percurso representado pelos pontos que representam cada referência após o mapeamento e por fim a ordem pela qual as referências serão visitadas, podemos ver um exemplo no anexo C.

Depois de se tomar a decisão quanto às heurísticas que seriam implementadas, pensou-se em experimentar aplicar as heurísticas por diferentes ordens e verificar qual seria a ordem das heurísticas melhorativas que daria melhores resultados, ou se essa ordem seria importante na qualidade dos resultados finais obtidos. Assim, testou-se para o armazém A de Frielas primeiramente apenas a heurística construtiva, posteriormente aplicou-se a heurística 1-optimal à solução obtida pela heurística construtiva, de seguida aplicou-se a heurística 2-optimal à solução obtida pela heurística construtiva, depois aplicou-se a 1-optimal seguida da 2-optimal e por fim trocou-se esta ordem, ou seja, primeiro a 2-optimal seguida da 1-optimal.

Apresenta-se na seguinte tabela os resultados obtidos para cada um dos processos acima descritos considerando trinta exemplos correspondentes a várias ordens de encomenda. Cada linha da tabela corresponde aos resultados obtidos para cada um dos exemplos.

Tabela 4.1: Testes da ordem das heurísticas

Construtiva		Construtiva_1opt		Construtiva_2opt		Construtiva_1opt_2opt		Construtiva_2opt_1opt	
Custo	Tempo	Custo	Tempo	Custo	Tempo	Custo	Tempo	Custo	Tempo
94	0.091	94	0.077	94	0.096	94	0.102	94	0.078
98	0.095	90	0.079	88	0.089	88	0.097	88	0.075
154	0.084	142	0.076	144	0.092	142	0.106	142	0.073
132	0.084	122	0.077	122	0.095	122	0.098	122	0.074
84	0.086	84	0.077	84	0.099	84	0.089	84	0.073
78	0.091	78	0.076	78	0.098	78	0.099	78	0.077
94	0.103	94	0.081	94	0.122	94	0.113	94	0.092
134	0.142	134	0.098	134	0.162	134	0.162	134	0.11
118	0.125	118	0.16	118	0.108	118	0.115	118	0.15
158	0.114	152	0.103	152	0.096	152	0.103	152	0.099
118	0.143	118	0.076	118	0.127	118	0.095	118	0.074
76	0.131	76	0.109	76	0.099	76	0.105	76	0.078
56	0.092	56	0.075	56	0.089	56	0.109	56	0.077
62	0.114	60	0.078	62	0.131	60	0.105	60	0.075
128	0.09	128	0.084	128	0.087	128	0.074	128	0.103
134	0.108	134	0.103	134	0.093	134	0.073	134	0.139
118	0.111	110	0.081	118	0.125	110	0.114	110	0.118
190	0.086	190	0.084	190	0.091	190	0.079	190	0.118
144	0.145	144	0.098	140	0.111	140	0.074	140	0.134
110	0.108	110	0.086	110	0.143	110	0.099	110	0.107
120	0.118	120	0.074	120	0.083	120	0.104	120	0.103
152	0.149	152	0.083	152	0.115	152	0.075	152	0.103
142	0.106	140	0.156	126	0.122	126	0.084	126	0.075
114	0.084	114	0.084	96	0.085	96	0.147	96	0.078
124	0.115	124	0.076	124	0.143	124	0.077	124	0.129
124	0.107	124	0.079	124	0.085	124	0.077	124	0.094
92	0.081	88	0.151	88	0.093	88	0.109	88	0.081
116	0.102	110	0.082	110	0.132	110	0.074	110	0.087
150	0.116	150	0.074	150	0.1	150	0.079	150	0.074
146	0.108	142	0.097	146	0.109	142	0.074	142	0.079
Média do tempo	0.1076	Média do tempo	0.0911	Média do tempo	0.1073	Média do tempo	0.097	Média do tempo	0.0942

Ao observar a tabela acima concluiu-se que quando se utiliza as duas heurísticas melhorativas, isto é, 1-optimal e 2-optimal, não houve quaisquer diferenças nos custos obtidos e no tempo de CPU, em segundos, não houve diferenças significativas. Notou-se também que houve oito casos em que os resultados obtidos através da utilização apenas da 1-optimal se diferenciaram dos resultados obtidos

através da utilização apenas da 2-optimal. Em quatro dos casos obteve-se melhor resultado ao usar apenas a 1-optimal e nos outros quatro casos obteve-se melhor resultado ao usar apenas a 2-optimal. Logo, não se pode tirar conclusões relevantes. Em termos de tempo computacional, os tempos são todos muito reduzidos.

De seguida, apresentam-se duas figuras que representam um exemplo de um desenho de uma rota antes e depois da aplicação das heurísticas, respetivamente.

Na Figura 4.1, podemos observar um exemplo de uma rota no armazém A de Frielas com doze pontos de *picking*. A rota é a seguinte (0, 41, 41, 46, 127, 167, 136, 132, 213, 213, 226, 304, 3, 0) e tem um custo de 290. As referências correspondentes são, respetivamente, “A01230209”, “A01240516”, “A02010412”, “A03230204”, “A04220204”, “A04030301”, “A04010305”, “A05230217”, “A05240305”, “A06050504”, “A08020417” e “A01010610”.



Figura 4.1: Exemplo de uma rota no armazém A de Frielas antes da aplicação das heurísticas

Na Figura4.1, para traçar a rota recorreu-se ao uso das cores, azul, laranja, verde e preto. A utilização das cores laranja, verde e preto será explicada a seguir.

Analisando a figura acima, facilmente se nota que se pode melhorar a rota. Assim que a rota começa, o primeiro percurso é ir de 0 para 41 (a preto). Ao longo deste percurso, passa-se pelo ponto 3, sendo que este também é um ponto de *picking* nesta rota, logo, não faz muito sentido não se recolher o produto que aqui se encontra, obrigando a uma segunda passagem por este ponto. Continuando com a análise, de seguida nota-se logo um cruzamento entre vários dos percursos da rota. Neste caso existem três percursos que se cruzam em grande parte dos mesmos. Para se distinguirem melhor recorreu-se ao uso das três cores laranja, verde e preto.

A experiência diz-nos que na obtenção de rotas, um dos problemas que torna o custo das rotas mais elevado é o cruzamento de percursos. No entanto nem sempre um cruzamento de percursos significa um pior resultado, isto é, um aumento do custo.

Na Figura 4.2, podemos observar o mesmo exemplo da Figura4.1. No entanto, a rota obtida nesta figura é diferente da rota obtida na figura anterior. A rota desta figura é a seguinte (0, 304, 226, 213, 213, 127, 41, 41, 167, 136, 132, 3, 46, 0) e tem um custo de 144. As referências correspondentes são, respetivamente, “A08020417”, “A06050504”, “A05240305”, “A05230217”, “A03230204”, “A01240516”, “A01230209”, “A04220204”, “A04030301”, “A04010305”, “A01010610” e “A02010412”.



Figura 4.2:Exemplo depois da aplicação das heurísticas

Nesta figura, os resultados obtidos são um pouco diferentes, aqui a rota parece mais circular e com menos cruzamentos de percursos. Nota-se um cruzamento no canto superior direito. Fazendo as contas verifica-se que a melhor opção em termos de custo corresponde à solução sem este cruzamento. O outro cruzamento que se nota é no canto inferior direito. No entanto, este cruzamento não interfere no valor final da solução, uma vez que quando se segue para o produto que se encontra no ponto 3, apanhar o produto que se encontra no ponto 46, antes ou depois do ponto 3 é indiferente porque teríamos sempre de voltar para o ponto 0 onde irá terminar a rota.

Com o objetivo de perceber em que medida é que as soluções obtidas pelas heurísticas implementadas diferiam das rotas atualmente efetuadas nos vários armazéns, compararam-se os custos das rotas realizadas em cada um dos armazéns antes da aplicação de qualquer heurística, após ser aplicada apenas a heurística construtiva e após serem aplicadas ambas as heurísticas melhorativas.

Para os armazéns B de Frielas, do Porto e de Braga, bem como para o armazém do piso 0 de Madrid, não foi possível comparar os resultados obtidos após aplicação das heurísticas com os resultados obtidos antes da aplicação das mesmas, uma vez que não havia informação sobre os mesmos.

Na tabela que se segue, podemos observar os resultados obtidos no armazém A de Frielas.

Tabela 4.2: Testes armazém A de Frielas

<b>Custo da rota</b>		
<b>Antes da aplicação das heurísticas</b>	<b>Após aplicação da heurística construtiva</b>	<b>Após aplicação de todas as heurísticas</b>
120	94	94
140	98	88
202	154	142
188	132	122
118	84	84
128	78	78
134	94	94
212	134	134
198	118	118
230	158	152

Observando os resultados obtidos para o armazém A de Frielas (Tabela 4.2), verifica-se que houve melhoramento após a aplicação de todas as heurísticas, em relação ao caso em que não se tinha aplicado nenhuma heurística (caso inicial), em 100% dos casos. A percentagem de melhoramento mínima foi no exemplo 1 e foi de aproximadamente 22%, a percentagem de melhoramento média foi de aproximadamente 33% e a percentagem de melhoramento máxima foi no exemplo 9 e foi de aproximadamente 40%.

Na tabela que se segue, podemos observar os resultados obtidos no armazém B de Frielas.

Tabela 4.3: Testes armazém B de Frielas

<b>Custo da rota</b>	
<b>Após aplicação da heurística construtiva</b>	<b>Após aplicação de todas as heurísticas</b>
148	148
163	163
165	159
179	177
99	99
125	121
167	141
135	135
179	177
137	123
112	112
192	165
118	118

Na tabela que se segue, podemos observar os resultados obtidos no armazém A do Porto.

Tabela 4.4: Testes armazém A do Porto

<b>Custo da rota</b>		
<b>Antes da aplicação das heurísticas</b>	<b>Após aplicação da heurística construtiva</b>	<b>Após aplicação de todas as heurísticas</b>
200	118	118
76	68	68
116	94	90
88	80	80
290	162	146
154	98	98
144	114	114
98	72	72
190	122	114
284	154	154



No caso do armazém A do Porto (Tabela 4.4), houve melhoramento após a aplicação de todas as heurísticas, em relação ao caso em que não se tinha aplicado nenhuma heurística (caso inicial), em 100% dos casos. A percentagem de melhoramento mínima foi no exemplo 4 e foi de aproximadamente 9%, a percentagem de melhoramento média foi de aproximadamente 30% e a percentagem de melhoramento máxima foi no exemplo 5 e foi de aproximadamente 50%.

Na tabela que se segue, podemos observar os resultados obtidos no armazém B do Porto.

Tabela 4.5: Testes armazém B do Porto

<b>Custo da rota</b>	
<b>Após aplicação da heurística construtiva</b>	<b>Após aplicação de todas as heurísticas</b>
73	73
101	101
95	95
93	93
105	105
45	45
134	134
159	159
125	125
119	119
99	99
134	134

Na tabela que se segue, podemos observar os resultados obtidos no armazém A de Braga.

Tabela 4.6: Testes armazém A de Braga

<b>Custo da rota</b>		
<b>Antes da aplicação das heurísticas</b>	<b>Após aplicação da heurística construtiva</b>	<b>Após aplicação de todas as heurísticas</b>
132	112	112
136	108	108
108	74	74
124	92	92
126	102	102
154	94	94
122	90	90
74	66	66
136	114	114
134	106	106

No armazém A de Braga (Tabela 4.6), houve melhoramento após a aplicação de todas as heurísticas, em relação ao caso em que não se tinha aplicado nenhuma heurística (caso inicial), em 100% dos casos. A percentagem de melhoramento mínima foi no exemplo 8 e foi de aproximadamente 11%, a percentagem de melhoramento média foi de aproximadamente 23% e a percentagem de melhoramento máxima foi no exemplo 6 e foi de aproximadamente 39%.

Na tabela que se segue, podemos observar os resultados obtidos no armazém B de Braga.

Tabela 4.7: Testes armazém B de Braga

<b>Custo da rota</b>	
<b>Após aplicação da heurística construtiva</b>	<b>Após aplicação de todas as heurísticas</b>
67	67
75	75
63	63
89	89
72	72
83	83
87	87
91	91
79	79
63	63

Na tabela que se segue, podemos observar os resultados obtidos no armazém do piso 0 de Madrid.

Tabela 4.8: Testes armazém do piso 0 de Madrid

<b>Custo da rota</b>	
<b>Após aplicação da heurística construtiva</b>	<b>Após aplicação de todas as heurísticas</b>
190	190
154	154
214	214
200	196
226	214
252	238
256	256
206	206
202	202
212	212

Após a análise das sete tabelas acima concluiu-se que a aplicação das heurísticas permitiu melhorar os resultados obtidos. Apesar da aplicação das heurísticas melhorativas não parecer tão importante, uma vez que a quantidade de casos apresentados em que houve efetivamente melhoria de resultados não foi assim tão grande, ao nível da empresa, a quantidade de rotas realizadas num só dia é muito maior do que o número que aqui foi apresentado. Se pensarmos ao nível semanal, ao nível mensal, ou ao nível anual, percebemos que a quantidade de rotas em que o resultado é efetivamente melhorado é enorme. Posto isto, as heurísticas melhorativas são importantes para a empresa porque no final de contas, o tempo poupado em rotas realizadas dentro do armazém é muito grande.

#### **4.2. Apresentação e análise dos resultados relativos ao problema de determinação de rotas numa zona de clientes afeta ao armazém de Frielas**

Para determinar as rotas de distribuição numa zona de clientes iremos ter um primeiro ficheiro de *input* que irá conter a geografia do armazém ao qual pertence a rota em questão, a zona na qual será feita a distribuição, o número total de encomendas prontas a expedir na zona e na hora em questão e ainda uma linha para cada pedido pronto a expedir que terá o número do cliente, o número da loja, o valor da encomenda e o dia e a hora a que foi feita a encomenda. De modo a calcular o tempo que passou desde uma determinada encomenda, utiliza-se uma função que nos dá a hora atual. Do ficheiro de *input* conseguimos retirar a hora a que a encomenda foi feita e depois é só fazer a diferença entre as duas horas e converter para a unidade de tempo que quisermos (horas, minutos, segundos, etc.), neste caso utilizou-se minutos. Tendo sempre em atenção se o pedido e a hora atual são no mesmo dia ou não, caso não sejam, isso também tem de ser contabilizado na contagem do tempo que passou. No anexo D encontra-se um exemplo desse ficheiro de *input*.

Como segundo ficheiro de *input* iremos ter a lista dos clientes pertencentes à geografia e à zona em questão. Nesse ficheiro teremos o número de clientes nessa zona dessa geografia, seguida de uma linha para cada cliente com o número do mesmo, o número da loja, o número do segmento ao qual pertence,

a hora de início de funcionamento desse cliente, a hora de fecho para almoço, a hora de abertura pós almoço, a hora de fecho de funcionamento, a hora de abertura ao sábado e a hora de fecho ao sábado. No anexo E encontra-se um exemplo desse ficheiro de *input*.

Teremos ainda um terceiro ficheiro de *input* que irá conter as distâncias entre todos os clientes da zona e da geografia em questão. Nesse ficheiro teremos duas linhas por cada cliente da zona e geografia e mais duas linhas para o depósito. Na primeira metade das linhas teremos uma linha para cada cliente e também uma para o depósito. Em cada uma dessas linhas teremos o número do cliente e o número da loja. No caso do depósito, o número do cliente é 0 e o número da loja também é 0. Na segunda metade das linhas, mantém-se uma linha por cliente e uma para o depósito. Em cada uma dessas linhas teremos a distância entre o cliente a que corresponde essa linha e todos os outros clientes e o depósito. No caso do depósito, será a distância entre o depósito e cada cliente. No anexo F encontra-se um exemplo desse ficheiro de *input*. Para a obtenção desse ficheiro foi necessário pegar nas coordenadas geográficas de todas as lojas de todos os clientes, e calcular a distância entre cada uma dessas lojas e também entre cada loja e o armazém (neste caso, o armazém de Frielas). Para calcular essas distâncias utilizou-se um programa que ao colocar as coordenadas das lojas e as coordenadas do armazém, nos dava como resultado o tempo, em minutos, e a distância, em quilómetros, entre cada par de coordenadas. No entanto, ao realizar este processo, notou-se que alguns destes valores não estavam corretos. De modo a corrigir os erros, identificaram-se quais os valores que pareciam mais fora do contexto e calcularam-se através do *Google Maps* estes valores. Para calcular através do *Google Maps*, colocaram-se as coordenadas uma a uma das lojas em questão e do armazém. Ou seja, em vez do *output* nos dar logo vários valores, estes valores eram obtidos individualmente.

Como ficheiro de *output* teremos um ficheiro que contém o número de pedidos que estavam prontos a expedir numa determinada zona e geografia antes do cálculo da rota, o número de clientes que correspondem a esses mesmos pedidos, uma vez que, se um cliente fizer dois pedidos a horas diferentes e entre estes pedidos não for realizada nenhuma rota, esses pedidos serão considerados na mesma rota. No entanto é preciso ter atenção ao facto de que ambos os pedidos correspondem ao mesmo cliente, isto é, podemos ter duas linhas diferentes que correspondem ao mesmo cliente correspondendo cada linha a cada um dos pedidos do cliente. Contém ainda qual a duração da rota em minutos e qual a rota a realizar. No anexo G encontra-se um exemplo desse ficheiro de *output*.

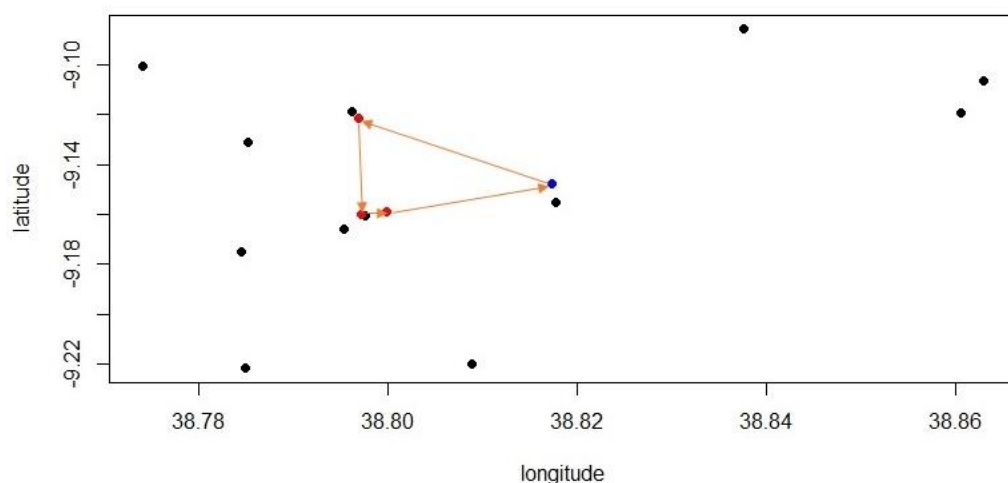


Figura 4.3: Exemplo de uma rota de distribuição

Na Figura 4.3, podemos observar um exemplo de uma rota de distribuição de uma dada zona e geografia, com 15 clientes, cada um com um só pedido. O tempo máximo desta rota é de 120 minutos. O ponto da figura com cor azul representa o depósito. A laranja podemos observar a rota realizada bem como os clientes que estão a ser servidos (a vermelho). Verifica-se que dos 15 pedidos no tempo de 120 minutos só 3 pedidos é que puderam ser satisfeitos. A rota foi de facto realizada em 120 minutos e o lucro das vendas foi de 167 €.

Geraram-se 25 exemplos considerando diferentes valores para o número de clientes e pedidos, e várias possibilidades de janelas temporais da parte da manhã e da tarde. Na tabela seguinte apresentam-se os resultados obtidos. Cada linha da tabela corresponde a um exemplo diferente.

Tabela 4.9: Testes efetuados para determinar as rotas numa zona de clientes

Número do Exemplo	Número de pedidos	Número de clientes	Número de clientes servidos	Duração da rota (em minutos)	Lucro das vendas (em €)
1	4	4	4	81	159
2	8	3	3	97	238
3	11	4	4	56	349
4	13	5	5	116	425
5	9	3	2	43	318
6	5	2	2	73	161
7	11	11	5	53	153
8	6	2	2	68	161
9	15	8	4	54	272
10	10	4	4	116	260
11	7	4	2	56	87
12	13	6	1	49	67
13	6	3	2	57	112
14	5	2	2	110	122
15	14	9	2	49	55
16	9	4	2	70	131
17	7	4	3	54	208
18	12	7	4	114	191
19	10	5	2	60	171
20	5	3	3	75	140
21	8	6	6	58	261
22	6	5	5	120	166
23	10	5	5	53	260
24	8	2	2	71	279
25	9	4	2	60	174

Nos exemplos 1 e todos os pares, o tempo máximo da rota é de 120 minutos. Nos exemplos ímpares, o tempo máximo da rota é de 60 minutos. Em alguns exemplos não foi possível visitar os clientes todos por incompatibilidade das janelas temporais ou por causa da duração máxima permitida para as rotas.

Pretendia-se durante o estágio comparar os resultados obtidos usando a heurística com as rotas atualmente efetuadas que são decididas pelos entregadores com base na sua larga experiência. Infelizmente tal não foi possível.

## Capítulo 5

### 5. Conclusões

Os objetivos deste estágio realizado na Norparts, uma empresa pertencente à Create Business foram a determinação de rotas de recolha de peças correspondentes a ordens de encomenda dentro do armazém (*order picking problem*) e a determinação de rotas de distribuição das peças pelos vários clientes de uma zona (*orienteering problem*).

Desenvolveram-se heurísticas construtivas em ambos os problemas e desenvolveu-se ainda uma heurística melhorativa para o *order picking problem*.

Para o *order picking problem* começou-se por fazer os mapeamentos dos armazéns necessários, utilizou-se o algoritmo de Floyd para determinar as distâncias entre os vários pontos do mapeamento e aplicou-se uma heurística construtiva, bem como duas heurísticas melhorativas.

Ao comparar os resultados obtidos após a aplicação das heurísticas com os resultados que estavam a ser obtidos antes da aplicação das mesmas, verificou-se que as rotas obtidas depois da aplicação das heurísticas foram sempre melhores .

Para o *orienteering problem* começou-se por tratar os dados de modo a obter as distâncias entre todos os clientes de uma zona e geografia e posteriormente desenvolveu-se uma heurística construtiva para obter as rotas de distribuição. Como trabalho futuro propõe-se a implementação de uma heurística de melhoramento, nomeadamente a heurística 2-optimal de modo semelhante ao que foi feito para o problema de determinação das rotas dentro do armazém.

Como trabalho futuro propõem-se também que sejam melhoradas as heurísticas já usadas, considerando também que se podem substituir ou adicionar novas.

No *order picking problem* sugere-se que se aumente o nível de detalhe do mapeamento de forma a que as rotas obtidos sejam ainda mais precisas.

No *orienteering problem* sugere-se que se obtenham os dados para determinar as rotas para todas as zonas de clientes e para todos os armazéns. Sugere-se também que se considere todos os clientes afetos a um armazém como um todo e não divididos por zonas e que a partir daí, com as viaturas disponíveis, se determine quais as rotas a realizar.





## **Bibliografia**

- [1] C. Theys O. Bräysy W. Dullaert and B. Raa. Using a TSP heuristic for routing order pickers in warehouses. *European Journal of Operational Research* 200: 755–763, (2010).
- [2] R. Daniels J. Rummel R. Schantz. A model for warehouse order picking. *European Journal of Operational Research* 105: 1-17, (1998).
- [3] L. Pansart N. Catusse H. Cambazard. Exact algorithms for the order picking problem. *Computers and Operations Research* 100: 117-127, (2018).
- [4] T. Chabot L. Coelho J. Renaud J. Côté. Mathematical Models, Heuristic and Exact Method for Order Picking in 3D-Narrow Aisles. *CIRRELT-2015-18*, (2015).
- [5] T. Chabot R. Lahyani L. Coelho J. Renaud. Order Picking Problems under Weight, Fragility and Category Constraints. *CIRRELT-2015-49*, (2015).
- [6] C. Archetti A. Hertz M. Speranza. Metaheuristics for the team orienteering problem. *J Heuristics* 13:49-76, (2007).
- [7] P. Vansteenwegen W. Souffriau D. Oudheusden. The orienteering problem: A survey. *European Journal of Operational Research* 209 1-10, (2011).
- [8] P. Toth D. Vigo. *Vehicle Routing: Problems, Methods, and Applications*. Second Edition. Bologna, (2014).
- [9] M. Gendreau A. Hertz G. Laporte M. Stan. A Generalized Insertion Heuristic for the Traveling Salesman Problem with Time Windows. *Operations Research* 46 (3): 330-335, (1998).
- [10] M. Gendreau A. Hertz G. Laporte M. Stan. New Insertion and Postoptimization Procedures Heuristic for the Traveling Salesman Problem. *Operations Research* 40 (6): 1086-1093, (1992).
- [11] H. Taha. *Operations Research: An Introduction*. Prentice Hall, 10 edition, (2017).
- [12] D. Rosenkrantz R. Stearns P. Lewis. An Analysis of Several Heuristics for the Traveling Salesman Problem. *Siam Journal on Computing*, 6, 3, (1977).
- [13] W. Cook W. Cunningham W. Pulleyblank A. Schrijver. *Combinatorial Optimizarion*. John Wiley & Sons, Inc. Capítulo 7, (1988).
- [14] R. Ramesh K. Brown. An efficient four-phase heuristic for the generalized orienteering problem. *Computers Ops. Res.* Vol. 18, No. 2, pp. 151-165, (1991).
- [15] C. Archetti D. Feillet A. Hertz MG. Speranza. The capacitated team orienteering and profitable tour problems. *Journal of the Operational Research Society* 60, 831-842, (2009).
- [16] Heurística. Acedido em 20 de Julho de 2020, em: <https://pt.wikipedia.org/wiki/Heur%C3%ADstica>
- [17] Como fazer a organização do armazém? Acedido em 22 de Julho de 2020, em: <https://www.fabrimetalarmazenagem.com.br/blog/organizacao-do-armazem/>
- [18] P. Black. Manhattan distance, in *Dictionary of Algorithms and Data Structures* [online]. Acedido em 25 de Julho de 2020, em: <https://www.nist.gov/dads/HTML/manhattanDistance.html>



## Anexos

### Anexo A - *input* do *order picking problem*

```
1010
7
A10140319
A03080201
A04090607
A06090216
A10050303
A02060424
A09240401
```

### Anexo B - output1 do *order picking problem*

```
Valor da solução: 118
Tempo: 0.125000
```

### Anexo C - output2 do *order picking problem*

```
Numero de pontos: 7
Custo da rota: 118
Percurso: 0 >> 449 >> 463 >> 435 >> 280 >> 192 >> 145 >> 97 >> 0
Ordem:
A10050303
A10140319
A09240401
A06090216
A04090607
A03080201
A02060424
```

### Anexo D - *input* dos pedidos do *orienteering problem*

```
1010
3
12
351 1 50 12 11 2020 19 6
351 1 14 4 11 2020 18 43
437 1 10 11 11 2020 0 52
437 1 11 7 11 2020 14 38
611 0 46 20 11 2020 0 7
667 1 17 2 11 2020 19 35
719 1 13 20 11 2020 13 28
719 1 23 12 11 2020 19 5
1152 0 25 11 11 2020 17 20
1152 0 17 2 11 2020 18 12
1152 0 18 4 11 2020 13 18
1208 1 23 15 11 2020 24 16
```

### Anexo E - *input* da lista dos clientes do *orienteering problem*

15								
122	0	1	420	750	850	1100	450	720
214	0	3	520	830	840	1100	500	750
296	0	1	530	810	820	1130	460	750
351	1	1	440	820	850	1050	490	810
437	1	1	470	760	820	1050	520	800
611	0	1	440	810	900	1050	440	800
667	1	2	460	790	890	1110	490	790
719	1	2	460	740	860	1020	440	750
1152	0	2	500	780	860	1110	470	730
1208	1	3	520	730	790	1050	440	820
1243	1	2	480	750	880	1040	470	820
1399	1	3	460	780	830	1100	480	720
1523	1	3	520	740	870	1060	500	780
1533	1	3	480	770	840	1130	420	770
1910	1	2	500	800	820	1040	540	790

### Anexo F - *input* das distâncias do *orienteering problem*

p	0														
122	0														
214	0														
296	0														
351	1														
437	1														
611	0														
667	1														
719	1														
1152	0														
1208	1														
1243	1														
1399	1														
1523	1														
1533	1														
1910	1														
0	29	20	28	24	26	28	27	27	16	30	24	20	17	24	19
26	0	16	30	30	30	29	18	25	26	24	25	26	19	30	25
23	19	0	23	29	22	16	24	27	22	25	26	23	28	27	21
21	30	28	0	20	23	23	18	24	17	23	22	16	22	23	18
15	25	23	24	0	25	15	29	27	19	18	18	18	21	28	30
20	27	23	22	27	0	27	23	19	26	21	28	26	15	30	17
19	29	25	15	24	28	0	24	29	20	26	21	15	30	22	24
16	27	21	15	16	20	17	0	16	19	25	17	30	18	20	16
27	28	21	16	25	25	19	22	0	30	18	18	28	25	24	21
25	26	15	22	29	25	29	20	21	0	26	22	30	30	29	24
19	20	15	29	26	26	27	24	17	27	0	30	21	26	19	23
18	23	27	26	28	16	27	28	17	17	27	0	28	23	15	27
30	16	26	25	19	15	29	21	17	28	28	20	0	21	30	29
25	24	16	22	25	29	15	18	19	19	27	17	23	0	18	16
27	18	15	24	27	17	21	19	29	28	26	23	21	22	0	18
24	30	27	30	28	24	30	29	30	20	27	16	18	21	28	0

### Anexo G - *output* do *orienteering problem*

```

Numero de pedidos: 12
Numero de clientes: 7
Duracao da rota: 114
Percurso: 0 >> 6 >> 1 >> 3 >> 2 >> 0

```